IDENTIFICATION OF MULTI-PART DIGITAL OBJECTS

A format for describing Representations

Jack O'Sullivan

Preservica Ltd

United Kingdom

jack.osullivan@preservica.com

0000-0002-0306-761X

Richard Smith

Preservica Ltd

United Kingdom

richard.smith@preservica.com

Jonathan Tilbury

Preservica Ltd

United Kingdom

jonathan.tilbury@preservica.com

Abstract – digital information often relies on the interactions of multiple individual digital files, rather than being completely encapsulated in a single file. Where this happens, all necessary files should be considered to be part of the same Multi-Part digital object. This paper describes a data model for describing the format of such digital objects, or more specifically, the Multi-Part Representations of such objects. As with the identification of file formats, having a Representation Format like this enables us to determine which tools to use to undertake particular preservation actions for a Representation.

Keywords – Format Identification, Data Model; Complex Objects

Conference Topics – Exploring the New Horizons; Scanning the New Development

I. BACKGROUND AND MOTIVATION

In a previous paper [1] describing Preservica's data model, we considered how digital files are somewhat arbitrary aggregations of digital data in the sense that they are artifacts of the implementation of data persistence, and not necessarily artifacts of the design of the digital data they encapsulate. This means that whilst a single file may (and often will) be an atomic unit of Digital Information, this is not guaranteed.

Digital information often relies upon the complex interaction of content from multiple distinct files. We may still think of this digital information as being a single digital object, composed of multiple "parts". Examples of such complex digital objects include emails, where the digital information requires both the message file and all attachments; captioned videos, where the captions may be provided in multiple languages, stored in multiple subtitle files alongside the video itself; and 3D object files, where the description of the shape of the object is held in a different file from the description of the surface materials required to render it.

Software to interpret and interact with such digital objects is going to require an understanding of what these files (parts) are and how they relate to

17th International Conference on Digital Preservation

iPRES 2021, Beijing,China.

DOI: 10.1145/nnnnnnn.nnnnnnn

each other, and this is as true of Digital Preservation systems and software as it is of any other type.

In this paper, we describe a schema for describing the format of such complex multi-part digital objects so that they can be identified, and so that relevant digital preservation tools, such as validators, property extractors and renderers can be registered for each type of object.

For precedent, we considered two cornerstone schemata in digital preservation and the identification of forms of digital content, the DROID Signature File [2] and the DROID Container Signature File [3]. Since the information on such formats would be stored in and delivered by Preservica's technical registry, we decided for consistency, and with a hope for future widespread adoption, to define the schema as a JSON Schema (draft-06) [4], in line with the entity definitions in the Preservation Action Registries (PAR) data model [5].

II. A Format for What Exactly?

Before determining how such a format should be constructed, we first had to determine exactly what we would be attempting to define a format for. There are various degrees of abstraction at which digital works can be considered, and there is a lot of overlap in how different types of work can be created and considered.

An illustrative example is to consider the digitization of a pre-existing physical book. A single book can exist in multiple physical forms, for example, hard or soft backed/bound, various physical page sizes, various type sizes etc; but the term "book" generally denotes information printed (or written) to multiple bound leaves. We may consider hard back and soft back to be different formats, but more likely we consider them the same format with different characteristics (e.g., the nature of the binding, the thickness or material of the backing). We may take different preservation steps depending on the characteristics of a given book, but we describe all books in the same basic way.

When we want to digitize a given book, we can store the resulting digital content in several different ways. We will typically scan an image of each page, typically creating a single digital file (typically TIFF or JPEG2000) per page. As long as we have some means of indexing these files so that they can be accessed in linear order (often achieved simply through a naming schema that includes a page number), we can stop at this stage and have a digitized book.

For added convenience, we can collate these images into a single digital file. This could be another image format allowing for multiple pages (TIFF for example), or another format entirely, such as PDF.

We now have two different forms of the same book, using two different types of file formats (TIFFs and JPEG-2000 are explicitly image formats, PDF is most often considered to be a document format), and two different numbers of files required. These both convey the same information (the original book), and so are both different ways of recording the same Intellectual Entity (PREMIS) or Information Object/Asset (Preservica). Both Preservica and PREMIS would refer to these as different Representations of the same information.

In order to make preservation decisions about these Representations, for example how to display or migrate them, we need to define the format that they are in. As such we will use the term Representation Formats to describe this new format entity.

The Representation is a mid-level abstraction of what the digital information actually contains; higher level than the technical details of the files involved, lower level than the abstract concept of "digitized book". However, it still needs the more technical view of "many image files", or "single document" to enable identification. This means characterizing the Representation is a higher-level conceptual task than characterising the individual files. As such, it is anticipated that this is a task that occurs during the ingest/pre-ingest process, after the initial file format identification.

III. Existing Format Definitions

The schema for a FileFormat in the DROID Signature File contains a lot of detail, but in essence it is comprised of 4 pieces of information:

1. An identifier for the FileFormat, used so that content can be unambiguously labelled as having that format;
2. Some descriptive metadata, a format name, version, date history, etc, context that can form part of the OAIS Representation Information;
3. Information about relationships to other FileFormats, typically prioritisation when a piece of content matches two different FileFormats;
4. A signature, a means of identifying that a piece of content has this FileFormat.

The signature is subdivided into ByteSequences which describe the sets of "magic" byte patterns that should be matched in the content, and any offsets for those patterns within the series of bytes that comprise the entire file.

The ContainerSignature extends this idea to allow us to specify that multiple "files" (FileStreams in PREMIS) within a single "container" file (e.g. a ZIP file), should be present, that each should be of a known internal name/path, and that each should have a specific format (described as a BinarySignature but more or less identical in nature to the InternalSignature of the FileFormat).

The same kind of information is required to define a Representation composed of multiple parts (a Multi-Part Representation); an identifier, some descriptive context, some idea of relationships to other Representation Formats, and a signature. The signature will need to be defined in terms of the numbers of types of files we are expecting, rather than explicit byte sequences, and we are unlikely to be able to make assumptions about things such as file names or paths.

The schema for this has been based on the work laid out by the PAR project, in part for consistency of entities in Preservica's Registry (which is PAR based), but also in part because we believe that this will enable us to extend the base PAR data model in the future so that the knowledge described and shared in PAR data can extend to working with complex, multi-part digital content.

In this initial work, we have been able to modify PAR Business Rules [6] in Preservica to reference these Multi-Part Representations. This has enabled us to define characterization and rendering of Multi-Part Representations in the same way as we do for characterizing and rendering individual files.

IV. Defining A New Representation Format

As discussed above, the Representation Format will need much the same information as a DROID FileFormat, thus there are parts of the schema that are very easy to define.

As a means of identification, we chose to use the PAR Identifier as the format for an "id" field, putting our Representation Format in an "extended PAR" namespace. This means we can record a name and namespace for each Format and provide a universally unique GUID for it.

As an initial set of descriptive metadata, we chose to allow the specification of a "name" element and a "description" element, both of which are free text entries. It is anticipated that this initial set may be expanded in time to allow information like versions, dates, publishers and other provenance information to be provided.

The most interesting part of the definition is of course, the RepresentationFormatSignature. This is defined as an array of criteria for the files comprising the multiple parts of the Representation.

Each "fileCriterion" can list an array of PAR Identifiers under the label of "formats", or an array of PAR Identifiers under the label of "formatFamilies", or both. These state that a particular file in a Representation matches the criteria for this Representation Format if it is of any of the formats listed in the "formats" array, or any of the formats associated with a format family in the "formatFamilies" array. Both the "formats" and "formatFamilies" arrays are optional, meaning both can be omitted. The semantic meaning of this is that the specified file is not limited to be identified as any particular format.

Alongside format requirements, there is a "minimum" and "maximum" to determine the number of files we expect matching that criterion. These are based on the minOccurs and maxOccurs

elements in XML schemas, allowing the value to be a number or in the case of “maximum”, the literal string “unbounded”. Both “minimum” and “maximum” are also optional, and the default for both is 1.

Taken together with optional format requirements, this means that the minimum possible definition of a file criterion is simply an empty JSON object, the semantic meaning of which is “exactly one file of any format”.

Example Signature

This format covers the “book digitized as an image per page” use case discussed earlier. For this we have a simple signature, represented as:

- fileCriteria:
  - formatFamilies:
    - “image”
  - minimum: 2
  - maximum: unbounded

This means a Multi-Part Representation composed of at least two files in any format Preservica recognises as an image will be recognised as this format.

Example 1 - Multi-Image Format Signature

This brief definition specifies the numbers and types of files expected but says nothing about any relative ordering of files. Ordering of content is likely to be important in several scenarios, for example it may determine how files in a Representation should be laid out or presented for display, but in most cases is essentially immaterial to identifying the format of the Representation.

As with file format identification, there is some cross-over between identifying content as being a particular format and asserting that it is a valid instance of that format. Where file numbers and formats are tightly prescribed, the identification may act as a de-facto validation, but in many other cases, validation will be an additional activity.

We identified early on that one very likely validation process would be to verify that all files expected are actually present. This is not necessarily possible in all cases, but in some, we know that multiple files are required in the Representation because they are referred to explicitly by other files in the Representation. The canonical example would be a Wavefront 3D Object, where an OBJ file (fmt/1210) explicitly references the materials (MTL files, fmt/1211) required for rendering.

In some cases, we may need to “prioritise” a particular file as being “primary”. The canonical example here would be an email file where attachments are referenced by name, but where an attachment may be another email.

A naïve definition of the EML signature might say “exactly one EML message file and an unbounded number of other files of any format”. To validate this, we might check that all files in the Representation are indeed referenced from the EML. If we have multiple EML files, without designating a specific ordering that asserts the “actual email in question” is the first we should process, it is ambiguous as to whether or not any other attachments are correctly referenced.

To deal with this scenario, the “representationFormatSignature” actually allows a “primaryFileCriteria” array as well as a “fileCriteria” array. The semantic meaning of this is that any “primaryFileCriteria” should match files *in the order that they are processed*.

Example Signature

This format covers "email" objects, where the object consists of the message file itself, and any attachments, stored as separate pieces of content. The signature is represented as:

- primaryFileCriteria:
  - formatFamilies:
    - "email"
- fileCriteria:
  - minimum: 0
  - maximum: unbounded

This means that we expect the first file processed in the Representation to be an email file, and that we allow any number of other files to be present. The way this signature is written, the "other" files are optional, meaning that a Representation composed of a single email file will also match this "Multi-Part Representation".

Example 2 - Email Format Signature

This does not prescribe how the files in the Representation should be ordered, merely that for some formats, the process order does matter. For example, in Preservica's data model, the list of Content Objects in a Representation is by definition an ordered list, so this ordering is used during representation format identification.

Finally, we defined a field named "hasPriorityOver", whose value is an array of other PAR Identifiers. This defines a one-way mapping of prioritization.

Example Representation Format

This format is a more-specific version of the Multi-Image format shown in Example 1. The signature is represented as:

- fileCriteria:
  - formatFamilies:
    - "renderable-image"
  - minimum: 2
  - maximum: unbounded

This means a Multi-Part Representation composed of at least two files in any format from the subset of all image formats that Preservica's rendering framework can display.

The overall format can be represented as:

- name: Renderable Multi-Image
- description: Multiple images, each of which is renderable
- signature: { as above }
- hasPriorityOver:
  - multi-image (Example 1)

Since any format that is part of the "renderable-image" family will also be part of the "image" family, any Representation that matches this format will also match the Multi-Image format from Example 1. The "hasPriorityOver" label here states that we should consider this as the higher priority format when assessing further actions such as how to display or migrate this Representation.

Example 3 - Renderable Multi-Image Representation Format

This can be used as a "tie-break" condition where matches to multiple Representation Formats are made, allowing a single format to be associated

DOI: 10.1145/nnnnnnn.nnnnnnn

with the representation. Alternatively, a digital preservation system may allow a representation to have multiple formats, storing each in an ordered list, and using them in that order to determine how to perform preservation actions. In that case, "hasPriorityOver" can be thought of as the ordering criteria for that list.

The class diagram required to describe a Representation Format is shown in Figure 1.

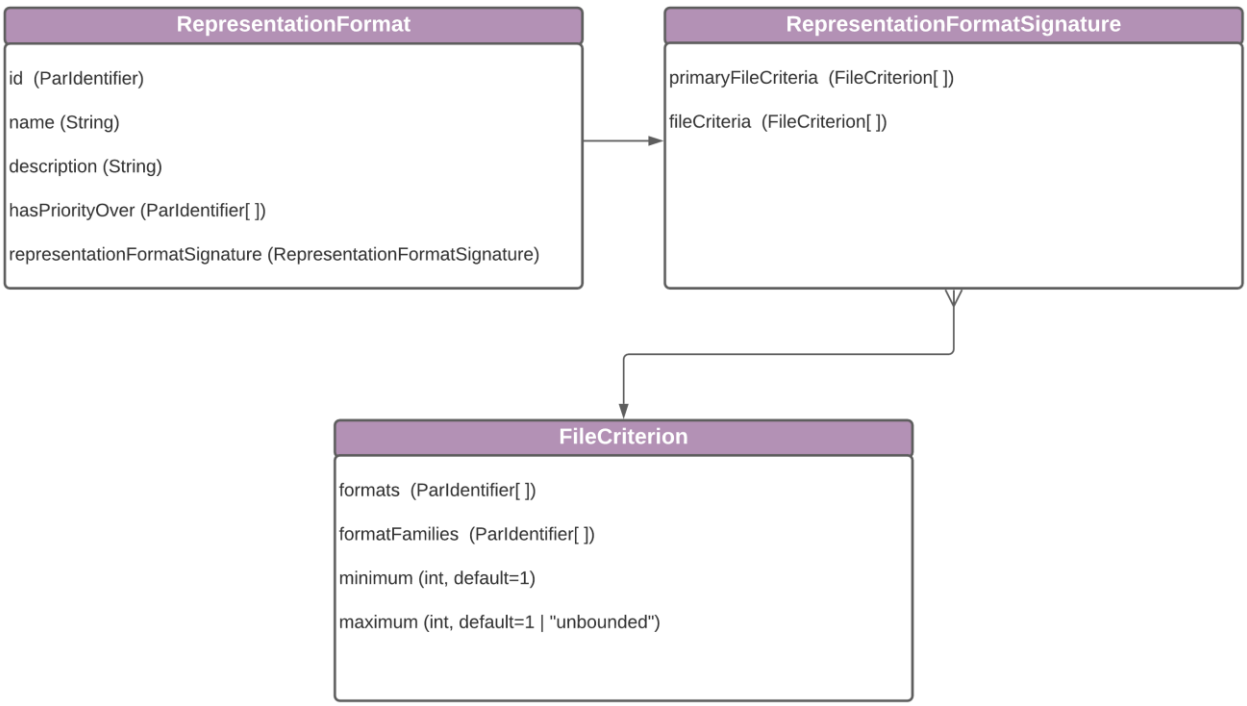

Figure 1 - Class diagram illustrating the Representation Format

V. FURTHER EXAMPLES

In order to test the correctness/validity of this format definition, we created numerous examples covering different uses of the signature, of which the examples in the previous section are a subset.

We have since also been able to use these as the basis for registering validation, property extraction and rendering tools explicitly against Representation Formats. Some further examples are described below.

A. Tweet Format

Tweets are the form of information created by, stored in, and displayed by Twitter. A single tweet is described by a JSON document (with an informal schema), which contains all relevant metadata about the text of the tweet, the user who sent it, the date it was sent, the location of the user at the time and so on. A tweet can also be posted along with up to four images, or a single video. We have observed that the images are always being retrieved as either JPEG or PNG, and the video is always retrieved in MP4 format, although this does not appear to be formally guaranteed and may just be "true at this time".

We have created a tweet Representation Format with a signature represented as:

- primaryFileCriteria:
  - formats:
    - JSON Tweet (fmt/1311)
- fileCriteria:
  - formatFamilies:
    - "jpeg"
    - "png"
  - minimum: 0
  - maximum: 4
- fileCriteria:

    - formats:
        - MP4 Video (fmt/199)
    - minimum: 0

This means that we expect the first file processed in the Representation to be a Tweet JSON file, then we expect 0 to 4 JPEG/PNG images and/or at most one MP4 video. In practice we will have either up to four images, or one MP4 video and one image (a thumbnail of the video). This is something that can be further checked with a specific validation tool.

B. Renderable Email Format

As with the Multi-Image example, we have a more specific format for the subset of email formats that Preservica can render, with a signature represented as:

- primaryFileCriteria:
    - formatFamilies:
        - "renderable-email"
- fileCriteria:
    - minimum: 0
    - maximum: unbounded

This means that we expect the first file processed in the Representation to be a renderable email file (in practice this means Internet Message Format fmt/278, or MIME Email fmt/950, but not others such as Microsoft Outlook Email Message x-fmt/430), and that we allow any number of other files to be present. The way this Representation Format is written, the "other" files are optional, meaning that a Representation composed of a single email file will also match this "Multi-Part Representation".

C. Captioned Video

Audio-Visual content is often an area where some Representations of the digital information require multiple files. We have concentrated on the example of a video file, encapsulating the video and audio streams, with closed-captions or subtitles being provided in separate files. This may be the case when the subtitles are provided as optional accessibility aids, and/or in multiple languages. We defined a signature represented as:

- fileCriteria:
    - formatFamilies:
        - "video"
- fileCriteria:
    - formats:
        - SubRip Subtitle File (fmt/1218)
    - minimum: 1
    - maximum: unbounded

This specifies that we are expecting a single file that Preservica recognises as being in a video file, and at least one subtitle file in SubRip Subtitle format. Because both criteria are simply defined as "fileCriteria", the ordering of the video and subtitle files within the Representation does not matter.

VI. CONCLUSIONS AND FUTURE WORK

We have described a simple format for identifying different types of Multi-Part Representations. This has been built on the precedents set by the various DROID/PRONOM definitions of a File Format and based on the JSON schema used to define PAR Core Entities, in the hope and anticipation that this will be useful to, and used by others.

There are several areas that we have identified as likely to need extension in future, however these needs are currently speculative.

The current set of descriptive metadata is limited to a title and description, which is likely to be insufficient over the long term.

The format definition allows you to specify expected files in terms of "formats" or "formatFamilies", but there are still some limitations to that. For example, we recognise that it may be necessary to describe a scenario where multiple files of the same family are permitted, but they must all be the same format within that family. In plain English, we might say "we expect two images, the images can be any image format, but must have

the same format as each other, i.e., two TIFFs or two JPEGs, but not one TIFF and one JPEG".

Similarly, as discussed briefly in the Tweet example, we cannot currently describe mutually exclusive sets of content where either is valid, but not both.

Despite its relative simplicity however, it has a flexibility that has proven to be powerful in defining formats for a number of disparate types of information.

Being based on the existing PAR Data Model has allowed us to extend our existing processes to deal with these types of Representation with relative ease.

PAR Business Rules provide a way of mapping a Preservation Action (which defines how a tool should run) to a list of formats [6]. This list is technically defined as a list of PAR Identifiers rather than full file format objects. So, although the original intention in the definition was that these formats were "file formats", by assigning PAR Identifiers to our Representation Formats, we have been able to map Business Rules to Representation Formats.

Preservation functionality in Preservica including characterization (property extraction and validation), migrations, and rendering, is driven by PAR Business Rules stored in Preservica's technical Registry. This has enabled us to define and implement those same types of preservation functionality for Representations and not just file formats.

Having this link means that functionality performed against a representation can be written to assume, or in the case of validation to assert, a particular set of content, rather than having to infer from one piece of content that other related pieces might be available if only we knew how to find it.

Based on the success of this initial work, we are planning on identifying and describing more types of Multi-Part Representations and extending Preservica's functionality to better understand and preserve the complex links between individual digital files.

REFERENCES

[1] J. O'Sullivan, R. Smith, A. Gairey and K. O'Farrelly, "A pragmatic application of PREMIS," in *iPRES 2019 Conference*, Amsterdam, 2019.

[2] A. Brown, "Automatic Format Identification Using PRONOM and DROID," The National Archives, London, 2006.

[3] The National Archives, "DROID 6.0 Technical Architecture," The National Archives, London, 2011.

[4] A. Wright and H. Andrews, "JSON Schema: A Media Type for Describing JSON Documents," 15 April 2017. [Online]. Available: https://tools.ietf.org/html/draft-wright-json-schema-01. [Accessed 29 April 2021].

[5] M. Addis, J. O'Sullivan, J. Simpson and J. Tilbury, "PAR Core Schema v0.1.0," 5 September 2019. [Online]. Available: https://github.com/openpreserve/par/tree/v0.1.0/schema. [Accessed 29 April 2021].

[6] M. Addis, J. O'Sullivan, J. Simpson, P. Stoke and J. Tilbury, "Digital preservation interoperability through preservation actions registries," in *iPres 2018 Conference*, Boston, 2018.